

CLAIMS:

- Sub A 81
- 1 A timer management system for managing timers in both a synchronous and asynchronous
2 system comprising:
3 an application program interface (API) providing a set of synchronous functions allowing an
4 application to functionally operate a timer;
5 a timer database for storing timer-related information; and
6 a timer services detecting the expiring of said timer, wherein a handle function of said timer
7 services allows said application to act on an expired timer without incurring an illegal time-out
8 message.
- 1 2. The timer management system as recited in claim 1, wherein said application performs the
2 following operations on said timer via said API:
3 creating said timer from an allocated block of system memory;
4 activating said timer; and
5 reinitializing said timer using said allocated block of system memory.
- 1 3. The timer management system as recited in claim 1, wherein said application performs the
2 following operation on said timer via said API:
3 creating said timer from an allocated block of system memory; and
4 activating said timer;

1 wherein said timer expires and said timer services sends synchronously a time-out message
2 to said application, wherein said time-out message is sent using said allocated block of system
3 memory.

1 4. The timer management system as recited in claim 1, wherein said application performs the
2 following operation on said timer via said API:

3 creating said timer from an allocated block of system memory; and
4 activating said timer;

5 wherein said timer expires and said timer services sends a time-out message to a particular
6 queue, wherein said timer is an expired state in an asynchronous state machine.

1 5. The timer management system as recited in claim 4, wherein said particular queue is a system
2 queue attached to said application.

1 6. The timer management system as recited in claim 4, wherein said application receives said
2 time-out message, wherein said handle function transfers said timer from said expired state in said
3 asynchronous state machine to an idle state in a synchronous state machine, wherein said handle
4 function allows said application to synchronously act on said timer.

1 7. The timer management system as recited in claim 4, wherein said application stops said
2 timer, wherein said timer is in an idle state in said asynchronous state machine with said time-out
3 message being queued.

1 8. The timer management system as recited in claim 7, wherein said time-out message is
2 dequeued, wherein said handle function transfers said timer from said idle state in said asynchronous
3 state machine to an idle state in a synchronous state machine, wherein said handle function allows
4 said application to synchronously act on said timer.

1 9. The timer management system as recited in claim 7, wherein said timer is deleted by said
2 application, wherein said timer is in a state in said asynchronous state machine in which said timer
3 is deleted and said time-out message is queued, wherein said time-out message is dequeued, wherein
4 said handle function transfers said timer from said state in said asynchronous state machine in which
5 said timer is deleted and said time-out message is queued to a non-existent state in a synchronous
6 state machine, wherein said handle function allows said application to synchronously act on said
7 timer.

1 10. The timer management system as recited in claim 7, wherein said timer is activated by said
2 application, wherein said timer is in a running state in said asynchronous state machine with said
3 time-out message being queued.

1 11. ⁹ The timer management system as recited in claim 10, wherein said timer is deleted by said
2 application, wherein said timer is in a state in said asynchronous state machine in which said timer
3 is deleted and said time-out message is queued, wherein said time-out message is dequeued, wherein
4 said handle function transfers said timer from said state in said asynchronous state machine in which
5 said timer is deleted and said time-out message is queued to a non-existent state in a synchronous

1 state machine, wherein said handle function allows said application to synchronously act on said
2 timer.

1 12. ¹² The timer management system as recited in claim 10, wherein said timer is stopped by said
2 application, wherein said timer is in said idle state in said asynchronous state machine with said
3 time-out message being queued, wherein said time-out message is dequeued, wherein said handle
4 function transfers said timer from said idle state in said asynchronous state machine to an idle state
5 in a synchronous state machine, wherein said handle function allows said application to
6 synchronously act on said timer

1 13. The timer management system as recited in claim 10, wherein said time-out message is
2 dequeued, wherein said handle function transfers said timer from said running state in said
3 asynchronous state machine to a running state in a synchronous state machine, wherein said handle
4 function allows said application to synchronously act on said timer.

1 14. The timer management system as recited in claim 4, wherein said application deletes said
2 timer, wherein said timer is in a state in said asynchronous state machine in which said timer is
3 deleted and said time-out message is queued, wherein said time-out message is dequeued, wherein
4 handle function transfers said timer from said state in said asynchronous state machine in which said
5 timer is deleted and said time-out message is queued to a non-existent state in a synchronous state
6 machine, wherein said handle function allows said application to synchronously act on said timer.

1 15. The timer management system as recited in claim 4, wherein said application activates said
2 timer, wherein said timer is in a running state in said asynchronous state machine with said time-out
3 message being queued.

1 16. The timer management system as recited in claim 15, wherein said timer is deleted by said
2 application, wherein said timer is in a state in said asynchronous state machine in which said timer
3 is deleted and said time-out message is queued, wherein said time-out message is dequeued, wherein
4 said handle function transfers said timer from said state in said asynchronous state machine in which
5 said timer is deleted and said time-out message is queued to a non-existent state in a synchronous
6 state machine, wherein said handle function allows said application to synchronously act on said
7 timer.

1 17. The timer management system as recited in claim 15, wherein said timer is stopped by said
2 application, wherein said timer is in an idle state in said asynchronous state machine with said
3 time-out message being queued, wherein said time-out message is dequeued, wherein said handle
4 function transfers said timer from said idle state in said asynchronous state machine to an idle state
5 in a synchronous state machine, wherein said handle function allows said application to
6 synchronously act on said timer.

1 18. The timer management system as recited in claim 15, wherein said time-out message is
2 dequeued, wherein said handle function transfers said timer from said running state in said
3 asynchronous state machine to a running state in a synchronous state machine, wherein said handle
4 function allows said application to synchronously act on said timer.

1 19. The timer management system as recited in claim 1, wherein said API is a DLL file.

[illegible]

1 20. A method for managing timers in both a synchronous and asynchronous system comprising
2 the steps of:

3 creating a timer from an allocated block of system memory by an application via an
4 application program interface (API);

5 activating said timer;

6 expiring of said timer; and

7 sending a time-out message to a particular queue when said timer expires, wherein said timer
8 is an expired state in an asynchronous state machine, wherein a handle function allows said
9 application to act on said expired timer without incurring an illegal time-out message.

1 21. The method as recited in claim 20, wherein said particular queue is a system queue attached
2 to said application.

1 22. The method as recited in claim 20 further comprising the step of:

2 receiving said time-out message by said application, wherein said handle function transfers
3 said timer from said expired state in said asynchronous state machine to an idle state in a
4 synchronous state machine, wherein said handle function allows said application to synchronously
5 act on said timer.

1 23. The method as recited in claim 20 further comprising the step of:

2 stopping said timer by said application, wherein said timer is in an idle state in said
3 asynchronous state machine with said time-out message being queued.

1 24. The method as recited in claim 23, wherein said time-out message is dequeued, wherein said
2 handle function transfers said timer from said idle state in said asynchronous state machine to an idle
3 state in a synchronous state machine, wherein said handle function allows said application to
4 synchronously act on said timer.

1 25. The method as recited in claim 23 further comprising the step of:
2 deleting said timer by said application, wherein said timer is in a state in said asynchronous
3 state machine in which said timer is deleted and said time-out message is queued, wherein said
4 time-out message is dequeued, wherein said handle function transfers said timer from said state in
5 said asynchronous state machine in which said timer is deleted and said time-out message is queued
6 to a non-existent state in a synchronous state machine, wherein said handle function allows said
7 application to synchronously act on said timer.

1 26. The method as recited in claim 23 further comprising the step of:
2 activating said timer by said application, wherein said timer is in a running state in said
3 asynchronous state machine with said time-out message being queued.

1 27. The method as recited in claim 26 further comprising the step of:
2 deleting said timer by said application, wherein said timer is in a state in said asynchronous
3 state machine in which said timer is deleted and said time-out message is queued, wherein said
4 time-out message is dequeued, wherein said handle function transfers said timer from said state in
5 said asynchronous state machine in which said timer is deleted and said time-out message is queued

1 to a non-existent state in a synchronous state machine, wherein said handle function allows said
2 application to synchronously act on said timer.

1 28. The method as recited in claim 26 further comprising the step of:

2 stopping said timer by said application, wherein said timer is in said idle state in said
3 asynchronous state machine with said time-out message being queued, wherein said time-out
4 message is dequeued, wherein said handle function transfers said timer from said idle state in said
5 asynchronous state machine to an idle state in a synchronous state machine, wherein said handle
6 function allows said application to synchronously act on said timer.

1 29. The method as recited in claim 26, wherein said time-out message is dequeued, wherein said
2 handle function transfers said timer from said running state in said asynchronous state machine to
3 a running state in a synchronous state machine, wherein said handle function allows said application
4 to synchronously act on said timer.

1 30. The method as recited in claim 20 further comprising the step of:

2 deleting said timer by said application, wherein said timer is in a state in said asynchronous
3 state machine in which said timer is deleted and said time-out message is queued, wherein said
4 time-out message is dequeued, wherein said handle function transfers said timer from said state in
5 said asynchronous state machine in which said timer is deleted and said time-out message is queued
6 to a non-existent state in a synchronous state machine, wherein said handle function allows said
7 application to synchronously act on said timer.

1 31. The method as recited in claim 20 further comprising the step of:
2 activating said timer by said application, wherein said timer is in a running state in said
3 asynchronous state machine with said time-out message being queued.

1 32. The method as recited in claim 31 further comprising the step of:
2 deleting said timer by said application, wherein said timer is in a state in said asynchronous
3 state machine in which said timer is deleted and said time-out message is queued, wherein said
4 time-out message is dequeued, wherein said handle function transfers said timer from said state in
5 said asynchronous state machine in which said timer is deleted and said time-out message is queued
6 to a non-existent state in a synchronous state machine, wherein said handle function allows said
7 application to synchronously act on said timer.

1 33. The method as recited in claim 31 further comprising the step of:
2 stopping said timer by said application, wherein said timer is in an idle state in said
3 asynchronous state machine with said time-out message being queued, wherein said time-out
4 message is dequeued, wherein said handle function transfers said timer from said idle state in said
5 asynchronous state machine to an idle state in a synchronous state machine, wherein said handle
6 function allows said application to synchronously act on said timer.

1 34. The method as recited in claim 31, wherein said time-out message is dequeued, wherein said
2 handle function transfers said timer from said running state in said asynchronous state machine to
3 a running state in a synchronous state machine, wherein said handle function allows said application
4 to synchronously act on said timer.